Technical Disclosure Commons

Defensive Publications Series

May 2021

RULES AUTOMATIC FINE-TUNING TO ENHANCE PROACTIVITY IN CUSTOMER SUPPORT USING RECURRENT NEURAL NETWORKS

HP INC

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation

INC, HP, "RULES AUTOMATIC FINE-TUNING TO ENHANCE PROACTIVITY IN CUSTOMER SUPPORT USING RECURRENT NEURAL NETWORKS", Technical Disclosure Commons, (May 11, 2021)
https://www.tdcommons.org/dpubs_series/4289

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

Title

Rules Automatic Fine-Tuning to Enhance Proactivity in Customer Support using Recurrent Neural Networks

Abstract

Proactive Customer Support is a support strategy that anticipates customer issues. It can either remedy them in advance or design service resources such that, when the failure does arise, the customer is presented with the solution without needing to contact support. Proactive Support offers several benefits for the business such as increase customer loyalty, decrease support calls, control communication, acquire new customers, retain existing customers, create advocates and protect against escalation. It also presents the opportunity to meet and exceed customer expectations, strengthen customer relationships, and boost the value of clients.

There are two types of customer help:

- Reactive support: Client calls support direct / indirect services to report the failure. Support team analyses the problem to resolve the issue as soon as possible.
- Proactive support: anticipates client issues and address them proactively (see Figure 1).

Proactive Support relies on several rules created and fine-tuned manually by experts. These rules are used to generate alerts that help support agents to proactively address customer issues. In LFP business, rules are defined using both threshold and time delta values. The former encodes the number of System Error (SE) occurrences (minimum frequency), and the latter the period (days) in which these SE should occur to be considered a relevant machine problem. More specifically, rules can be categorized as PRO (well-defined rules, no fine-tuning is needed) or MONITORING (newly defined rules, fine-tuning is highly required). All rules are implemented in a rule engine named SDS. Once a rule is satisfied, an alarm is created in the Smart Service Console (SSC). Afterwards, the agent opens a new case in CDAX database to store information related to the printer issue (*e.g.*, open / close dates, category, comments). The case is closed when the machine is fully repaired (see Figure 1).

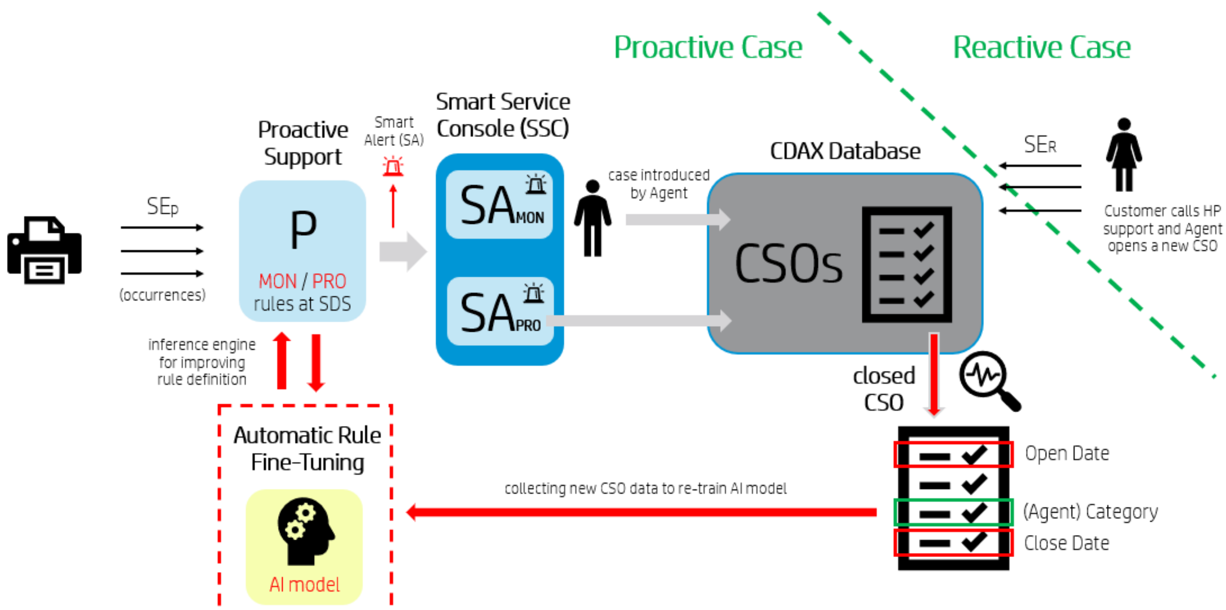

Figure 1: Proactive Support workflow: rules are currently defined, reviewed and deployed manually by operators. To improve the overall pipeline, a new module named "Automatic Rule Fine-Tuning" is proposed to optimize rule fine-tuning.

The methodology presented herein provides valuable information on which rules need to be fine-tuned as they are not working properly (too many reactive cases or alarms triggered). We develop a system to accurately analyse time series data and SE patterns and find the optimal parameters (*i.e.*, threshold, time delta) for each rule. Our approach generalizes well for a wide variety of SE, printers and programs without increasing model complexity. Recurrent Neural Networks (RNN) are employed as they are proven to be efficient in handling time series data.

Rule deprecation is also covered when new thresholds and time deltas do not improve the performance of the original rule definition.

Problem Solved

This invention allows to fine-tune existing rules from different programs by addressing the following challenges:

- To identify which specific rules need to be fine-tuned by using customer feedback (reactive calls).
- To overcome expensive manually designed rules while minimizing human errors, time and costs.
- To design an end-to-end RNN that automatically computes the optimal threshold and time delta.
- To improve efficiency and scalability of our HP proactive support system.

Prior Solutions

The current LFP do not include a solution to this problem. To the best of our knowledge, no rule automatic adjustment algorithm has been implemented / deployed yet to enhance proactivity in customer support. Today, the SE identification, rule design, validation and fine-tuning are carried out manually by an operator, which is time-consuming, error prone, not scalable across multiple programs and require a large amount of resources. Therefore, a robust statistical-based methodology to compute the exact performance (*e.g.*, precision, recall) for each rule is strongly encouraged to improve our customer support and service while reducing the associated human costs (manual intervention) for the business.

Description

<u>Data Sources</u>

Three data sources were considered to train / test our approach. Data collected was from July'19 to December'20.

- Printer Telemetry data (extracted from SEALS database): Daily occurrences of the events (rules) for all product and serial number combinations. Data used is agnostic to firmware version and updates.
- CSO data (extracted from CDAX database): Customer service calls reported for all product and serial number combinations. Cases include CSO open and close dates, system errors, category, among others.
- Rules data (extracted from SSC): Event codes include status (*e.g.*, PRO, MONITORING), time delta and threshold values. Top 100 rules were chosen to firstly assess the proposed methodology.

<u>Data Preparation</u>

The rule optimization procedure involves the computation of suitable threshold and time delta pairs by processing SE information as time series data. This sequential data is labelled as failure or non-failure using CSO information (*i.e.*, open / close dates), and prepared through a sliding window approach based on day-wise SE occurrences. The window size (10 days) was selected experimentally. To label the windows as failure (class 1) or non-failure (class 0), a hyperparameter 'X' (range of days for which a CSO is assumed to be reported due to SE occurring in a window) is defined. Windows are classified as 1 (failure) if a CSO got reported till 'X' days from the window end. Otherwise, the window is classified as 0 (non-failure). 'X' was also selected experimentally (5 or 10 days).

<u>Model Architecture</u>

The designed deep learning architecture is based on a RNN with attention mechanisms (see Figure 2). The gated unit used in the current solution is GRU (*a.k.a.* gated recurrent unit), since it improves the memory capacity, reduces the model complexity, and consequently, speeds up the network training on less data. This hidden unit is also applied for settling the vanishing gradient problem. The GRU input are the failure and non-failure labelled windows, and the learning model is defined as a binary classification. The final output class is calculated through a dense (sigmoid) layer.

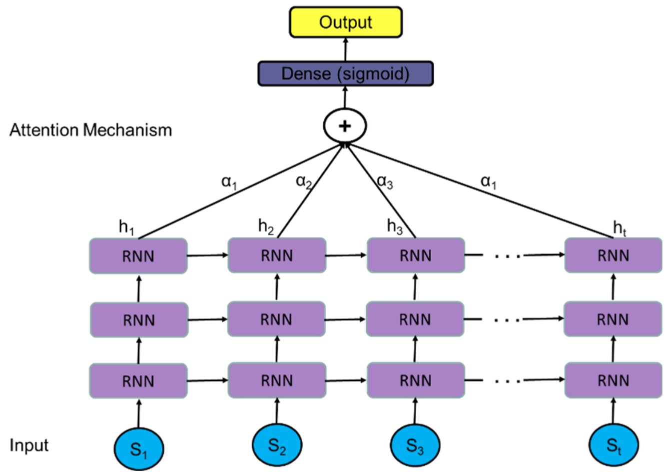

Figure 2: Recurrent neural network architecture with attention mechanisms.

Attention mechanisms allow to focus on important parts of the SE input sequences (windows) when predicting optimal threshold and time delta pairs, which enables easier learning and of higher quality. They let our model to know exactly the last SE (contained in a specific day) that will trigger the alarm. More specifically, learned attention weights (or probability distribution of event days) from correctly classified datapoints are utilized to identify the best combinations of index (window timestamp) and SE frequency. These values will represent the fine-tuned time delta and threshold for each rule. They are thus computed following the steps described below:

1. For each fold, the epoch with the lowest test loss (*i.e.*, the exact moment when the RNN is predicting better) is picked to compute the time delta and threshold.
2. For this specific epoch, attention values are analyzed to identify the best position (index) in which the maximum attention provides a correct classification.
3. All attention positions are grouped. For each attention index, the minimum cumulative frequency across datapoints is chosen.
    a. The minimum cumulative frequency value becomes the threshold.
    b. The position (index) value becomes the time delta.
4. For all pairs of time delta – threshold, the precision and recall are computed internally and compared with the original rule definition. If both performance measures are higher, the new pair is proposed by the network. The other pairs (with lower precision and recall) are thus automatically discarded.

To assess the proposed approach, we have estimated how many (past) reactive cases could have become proactive if the new adjusted rule had been applied.

Advantages

Compare to the current manual procedure, this invention provides the following new inputs:

- Intelligent, precise and fast rule fine-tuning for an efficient HP customer support and service.
- Fully automatic and flexible DL method to adjust rules across multiple programs. Our algorithm is also capable to learn SE patterns and recognize outliers over time, and consequently, to ensure an effective rule behavior.
- Optimal rule parameters (*i.e.*, threshold, time delta) are found without manual intervention. Thus, human errors made by operators are drastically reduced.
- Time, costs and personal resources are lowered. Scalability is highly improved when many rules need to be fine-tuned.

Disclosed by Jordina Torrents-Barrena, Prateek Kushwaha, Vivek Tiwari, Neha Bansal, HP Inc.